What Should We Forecast? Benchmarking Agents on Early Question Discovery

Keisuke Ueda [1] **Robert West** [1] **Veniamin Veselovsky** [2]

Abstract

Forecasting systems are usually evaluated after someone has already specified the question. HorizonBench evaluates the preceding step: whether an agent can identify which future uncertainties are worth forecasting before they become salient. We construct a set of later-resolved events from future news, give agents only cutoff-bounded historical news, and score whether their generated questions cover the same underlying uncertainties. Across 445 geopolitical events, frontier agents recover only a low-single-digit fraction of future-salient events at fixed candidate budgets, with substantial variation across models and event types. These results suggest that early question discovery remains a distinct bottleneck for AI forecasting.

1. Introduction

> *There are known knowns, things we know that we know; and there are known unknowns, things that we know we don't know. But there are also unknown unknowns, things we do not know we don't know.* — Donald Rumsfeld

Forecasting systems are usually evaluated on known unknowns: given a well-specified future event, assign a calibrated probability. But many real-world foresight failures happen one step earlier. The important question is not misforecasted; it is never asked until the event is already underway. As AI systems become increasingly competitive at forecasting well-specified questions (Zou et al., 2022; Halawi et al., 2024; Karger et al., 2025; Alur et al., 2025; Metaculus, 2026), this assumption becomes more important to examine. Existing forecasting benchmarks usually provide the question in advance. They therefore evaluate probability assignment conditional on a known uncertainty, while leaving out the upstream problem of discovering which uncertainties should be tracked in the first place.

In hindsight, many crises have visible precursors. The hard problem is not merely interpreting strong evidence; it is surfacing the right uncertainty while the evidence is still sparse, noisy, and easy to ignore. We introduce HorizonBench, a benchmark for evaluating early question discovery — how well can an AI system detect operational unknown unknowns. Given only news available before a cutoff date, agents must generate forecasting-style questions about future events that may later become salient. We then measure whether those questions cover surprising resolved events constructed from subsequent news, and how far in advance they are surfaced.

HorizonBench has two sides. On the future side, we construct a pool of later-salient resolved events from news in a target period. On the past side, agents receive only pre-cutoff news and generate candidate forecasting questions. An offline judge then matches generated questions to temporally eligible events. This design evaluates automated question-generation systems on whether their ex ante questions cover the events that later matter, without giving them those events in advance.

We evaluate multiple frontier models across regions and lead-time windows, measuring event-level coverage, question efficiency, lead time, and a news-volume surprisingness metric. We find that early question discovery is difficult even for frontier agents: at fixed candidate budgets, models cover only a small fraction of later-salient events, and GPT-5.5 is consistently strongest but still far from exhaustive. Performance is not explained by verbosity alone, and agents struggle especially when an event's post-resolution news volume rises sharply relative to its pre-resolution footprint. HorizonBench therefore isolates a capability that is upstream of probability calibration: discovering the right uncertainties early enough to matter.

2. Related Work

Forecasting benchmarks. Recent AI forecasting benchmarks and tournaments now test increasingly agentic fore-

[1]EPFL [2]Preseen. Correspondence to: Keisuke Ueda <keisuke.ueda@epfl.ch>.

Forecasting as a New Frontier of Intelligence. Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

casters across news, markets, and open-ended settings (Zou et al., 2022; Halawi et al., 2024; Karger et al., 2025; Ye et al., 2024; Zeng et al., 2025; Yang et al., 2025; Wildman et al., 2025; Liptay et al., 2026; Yuan et al., 2025; Tao et al., 2025; Cheng et al., 2026; Bianchi et al., 2025; Bosse et al., 2026), including systems built with reinforcement learning, scaffolding, simulation, and industry pipelines (Turtel et al., 2025; Chandak et al., 2026; Alur et al., 2025; Mühlbacher et al., 2024; Bosse et al., 2025; Kazinnik & Sinclair, 2025; Hansen et al., 2025; Jeen et al., 2026; Turtel et al., 2026a;b). HorizonBench keeps the time-bounded setup but moves the task upstream: the model must decide which forecastable questions should exist before the relevant event is given.

Prediction markets also evaluate forecasts on pre-specified questions, using financial incentives to aggregate beliefs about future outcomes (Wolfers & Zitzewitz, 2004), and can provide useful early signals for economic forecasting (Diercks et al., 2026; Kalshi, 2026).

Question generation and open-ended prediction. Automated pipelines generate forecasting questions from text, markets, or future-event databases (Bosse et al., 2026; Karger et al., 2025; Bianchi et al., 2025; Yuan et al., 2025; Tao et al., 2025; Cheng et al., 2026), often to scale training and reinforcement-learning data for forecasting agents (Turtel et al., 2025; Chandak et al., 2026; Turtel et al., 2026a;b). Open-ended prediction benchmarks similarly ask models to forecast outcomes or timelines from a topic, background, or question (Guan et al., 2024; Ye et al., 2024; Zeng et al., 2025; Yuan et al., 2025; Tao et al., 2025; Bianchi et al., 2025). These efforts expand forecasting beyond static question sets, but they usually condition on a visible topic, seed article, event background, or already-written question. HorizonBench instead starts from later-salient resolved events, reconstructs the pre-event information environment, and asks whether an agent could have surfaced the underlying uncertainty without being given it as a seed.

Weak signals. Weak-signal mining studies precursors, anomalies, and underappreciated evidence in historical data (Ansoff, 1975; Ebadi et al., 2022; Zhang & Evans, 2025; Turtel et al., 2026a;b). Recent AI foresight work uses LLMs, retrieval models, and event/causal graphs to accelerate horizon scanning and weak-signal discovery (Ashouri et al., 2026; Schmidt et al., 2025; O'Keefe & Marshall, 2025; Rozanec et al., 2024).

3. Methodology

3.1. Evaluation Target: Early Question Discovery

Standard forecasting evaluates probability assignment on a known question q. HorizonBench evaluates the preceding step. An evaluation instance specifies an area, a cutoff t, and a target window $T = [t_s, t_e)$. At cutoff t, the agent sees only a time-bounded information environment $\mathcal{D}_{\leq t}$ and emits a ranked set of future-facing questions $Q_\theta(t, T) = \{q_1, \ldots, q_k\}$, intended to identify uncertainties that may resolve inside T.

After the target window closes, we construct the corresponding pool of salient resolved events $C_T = \{c_1, \ldots, c_n\}$, where each $c \in C_T$ resolved inside the same target window T and area used for question generation. Each event is represented as a structured record containing its canonical uncertainty, resolution, accepted phrasings, rejected near-misses, and hindsight signals; Appendix C shows the record schema and judge rubric, and Appendix B summarizes the full construction pipeline. Let $m(q, c) = 1$ when the judge finds that generated question q captures the same underlying uncertainty as event record c. For the paired question set $Q_\theta(t, T)$ and event pool C_T, coverage is $|\{c \in C_T : \exists q \in Q_\theta(t, T), m(q, c) = 1\}|/|C_T|$, and precision is $|\{q \in Q_\theta(t, T) : \exists c \in C_T, m(q, c) = 1\}|/|Q_\theta(t, T)|$. The task is not ordinary retrieval: the future event may not yet have a name, and the agent must decide which weak patterns are worth turning into forecastable questions.

3.2. Pipeline

HorizonBench is *target-window anchored*: each evaluation instance is keyed by area, lead band, target window, cutoff, and model. Resolved events serve as a matching pool rather than per-question anchors; an event is eligible iff its subarea maps to the run's area and its resolution date falls inside the half-open target window $[t_s, t_e)$.

Future-side event construction. For each area, models search news from the target period and list consequential uncertainties that resolved during that period. We run five rounds. In later rounds, the model sees events already found and is asked to find different ones, which broadens the pool beyond the most obvious events. We then cluster duplicates and convert each unique event into a structured record with accepted phrasings, rejected near-misses, hindsight signals, and resolution metadata (Appendix B).

Past-side question generation. For each model, area, target window, and lead band, the agent is restricted to pre-cutoff news through a cutoff-safe Perigon[1] wrapper that enforces half-open UTC query windows and clamps every end date to the cutoff. Generation is repeated cumulatively: later rounds see earlier questions and are asked to add non-duplicate candidates.

Offline window-batch judge. Matching is a single offline

[1]Perigon is a commercial news API that supports article search over title, description, and content, with date, source, language, and other metadata filters.

Model	Lead time	Coverage@K			Precision@K		
		K=10	K=20	K=50	K=10	K=20	K=50
GPT-5.5 (medium)	7–14 days	2.25	2.92	5.62	13.75	8.75	6.50
	14–28 days	2.25	3.15	5.62	25.00	20.00	13.50
Claude Sonnet 4.6 (max)	7–14 days	2.02	2.92	3.82	12.50	9.38	6.25
	14–28 days	1.12	2.25	2.92	12.50	13.75	8.50
DeepSeek v4 Pro (xhigh)	7–14 days	1.80	2.92	3.82	10.00	8.75	5.00
	14–28 days	1.57	2.25	2.92	17.50	13.75	7.00
Gemini 3.1 Pro (xhigh)	7–14 days	2.25	2.70	3.60	12.50	8.12	4.50
	14–28 days	1.80	2.02	2.92	22.50	12.50	8.00

Table 1. Coverage@K and Precision@K by model and lead time, pooled across areas. Coverage values use the 445-event pool as denominator; precision is candidate-level micro precision. Values are percentages rounded to two decimals.

LLM-as-a-judge pass per evaluation instance: the merged candidate list and all eligible events for the window are scored in one batched call with no live tools. The judge is polarity-neutral, so events that resolved NO remain detectable. It emits sparse candidate–event match pairs, from which event-level hits are derived.

3.3. Experimental Setup

Data and windows. We evaluate Middle East / North Africa (MENA) and Eastern Europe / Russia (EER) over April 1–28, 2026. The future-side pipeline yields 445 resolved events (229 MENA, 216 EER). Six target windows are paired with fixed cutoffs: four one-week windows at ≈one-week lead (d7_14) and two two-week windows at ≈two-week lead (d14_28).

Models and metrics. We evaluate GPT-5.5, Claude Sonnet 4.6, DeepSeek v4 Pro, and Gemini 3.1 Pro Preview. Across two areas, six target windows, and four models, the offline matcher scores 48 merged candidate sets. Perigon's article API supports Boolean search over title, description, and content, along with publication-date, language, and source filters (Perigon, 2026c;a;d;b). We report fixed-budget coverage, candidate-level precision, and earliest-finder attribution. Because useful but unrealized questions are unlabeled, precision should be read as a labeled-event precision proxy.

Surprisingness. For each event, a model selects five 3-noun anchor sets, such as {US, Iran, attack}. For each set, we use Perigon to count articles matching the Boolean conjunction of the three nouns in matched 7-day pre- and post-resolution windows. The per-set score is $\log((n_{\text{post}}+1)/(n_{\text{pre}}+1))$, and event surprisingness is the average of the five scores.

4. Results

4.1. Quantitative results

Coverage and precision@K. Table 1 reports the main fixed-budget results. We first ask whether agents cover later-resolved events under equal candidate budgets. Coverage@K is the fraction of the 445-event pool hit by the union of top-K generated questions across pooled areas within each lead band. Precision@K is candidate-level micro precision: the fraction of top-K submitted candidates that match at least one eligible event.

The low absolute coverage is the central result. Even with 50 ranked questions per candidate set, the best model covers only 5.62% of the 445-event pool in each lead band, indicating that current agents leave most later-salient uncertainties undiscovered. Because the comparison is fixed-budget, this result is not an artifact of one model submitting longer lists.

Precision decreases as K grows, as expected from ranked candidate lists. GPT-5.5 has the strongest combination of coverage and precision at $K = 50$, especially in the 14–28 day band, where it reaches 13.50% Precision@50.

The table should be read as an equal-budget comparison: every model receives the same top-K candidate budget within each model-area-window run. Thus the low coverage is not an artifact of one model submitting fewer questions, and the precision drop with K quantifies the cost of expanding the question list.

Full top-K coverage and precision curves, along with a secondary comparison to GDPval-AA Elo, are included in Appendix F.

Earliest-finder win rate. Coverage does not show which model surfaced a shared hit first. We therefore assign each event hit by at least one model to the model that found it at the earliest eligible cutoff, splitting credit for ties. Under the equal-budget top-50 evaluation, 54 events are hit by the four-model panel; GPT-5.5 receives 50.8% of fractional

Model	Win credit	Share (%)
GPT-5.5 medium	27.42	50.8
Claude Sonnet 4.6	9.42	17.4
DeepSeek v4 Pro	9.08	16.8
Gemini 3.1 Pro	8.08	15.0

Table 2. Earliest-finder win rate over the 54 events hit by at least one model under the equal-budget top-50 evaluation. Credit is split evenly across tied earliest models; shares are percentages.

Event category	Events	Any-model hit (%)
Sanctions / restrictions / coercive pressure	42	21.4
Domestic politics / governance / leadership	51	19.6
Economy / energy / trade / infrastructure	31	16.1
Diplomacy / alliances / conflict settlement	87	12.6
Humanitarian / migration / civil society	17	11.8
Kinetic conflict / acute security incidents	100	10.0
Defense posture / capability / assistance	70	8.6
Law / courts / rights / accountability	27	3.7
Covert action / intelligence / cyber / sabotage	20	0.0

Table 3. Any-model hit rate by event category. Values are percentages over the 445-event pool.

earliest-finder credit, while the other three models cluster between 15–17% (Table 2); the by-area decomposition is in Appendix Table 11.

This result strengthens the main coverage finding: GPT-5.5 is not only the strongest model by fixed-budget coverage, but also most often the first model to surface an event when the panel finds it at all. The remaining models are much closer to each other than to GPT-5.5 under this attribution rule.

4.2. Analysis

Event categories. We categorize all 445 resolved events using a mechanism-first taxonomy, assigning each event to exactly one primary category according to how the event resolves. We then cross-tabulate these categories with the equal-budget K=50 hit set. Table 3 shows that misses are broad-based, with every category below 22% recall, but it also reveals sharper blind spots in law/accountability and covert/cyber events. Sanctions and domestic politics are relatively easier for agents to surface, while law/accountability and covert/cyber events are systematically missed.

Surprising events. We quantify event surprisingness with the Perigon log-ratio metric defined in Section 3.3. Figure 1 shows that the event pool is concentrated near modest positive shifts in coverage, while any-model hit rates are highest for the least surprising bins and generally lower for high-surprisingness events. This suggests that current agents are better at surfacing issues that already have a visible pre-event footprint than events whose coverage increases sharply only after resolution.

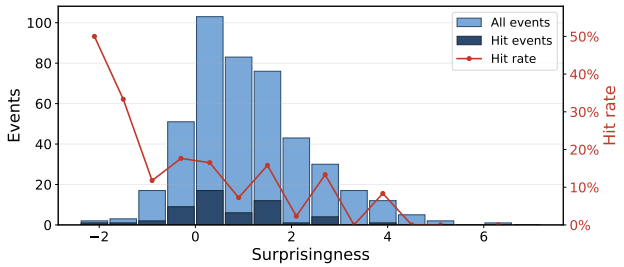

Figure 1. Surprisingness and discovery rate. Bars show the distribution of 445 events by Perigon-based surprisingness score; dark bars indicate events hit by at least one model, and the red line shows the any-model hit rate per bin. Surprisingness is the average log post/pre article-count ratio over five 3-noun anchor queries.

Together, the category and surprisingness analyses point to the same conclusion: the bottleneck is not simply that agents need to generate longer lists. Many missed events require selecting the right mechanism, instance, or weak local signal before the event becomes broadly visible.

5. Discussion and Limitations

HorizonBench separates two capabilities that are often conflated: forecasting a specified question and deciding which questions should exist. Our results suggest that early question discovery remains a bottleneck for frontier agents. Models recover only a small fraction of later-salient events, and their misses are especially pronounced for events whose pre-resolution news footprint is weak.

For automated question-generation and forecasting-data pipelines, and for prediction-market platforms such as Kalshi and Polymarket, the relevant target is not merely plausible questions tied to visible topics; it is coverage of important events that later resolve. HorizonBench approximates this by comparing ex-ante generated questions against a later-resolved event set, making missed future coverage visible.

We are also building a forward-testing version of HorizonBench: agents will submit questions before the target window, the submissions will be frozen, and coverage will be scored only after the window resolves. This removes retrospective event selection from the evaluation loop while preserving the same question-discovery target.

The main limitations are construct and scope validity. HorizonBench scores only events that became salient ex post, so useful questions about unrealized but genuinely important risks may receive no credit; this also means our precision metric is a labeled-event proxy rather than full precision. The event set and matching labels are model-assisted and news-dependent, so they may inherit source, event-selection, and judge biases.

References

Alur, R., Stadie, B. C., Kang, D., Chen, R., McManus, M., Rickert, M., Lee, T., Federici, M., Zhu, R., Fogerty, D., Williamson, H., Lozinski, N., Linsky, A., and Sekhon, J. S. AIA Forecaster: Technical report, 2025. URL https://arxiv.org/abs/2511.07678.

Ansoff, H. I. Managing strategic surprise by response to weak signals. *California Management Review*, 18(2): 21–33, 1975. doi: 10.2307/41164635.

Artificial Analysis. GDPval-AA: Agentic real-world work-task evaluation. https://artificialanalysis.ai/evaluations/gdpval-aa, 2026. Accessed: 2026-05-13.

Ashouri, S., Cole, C., Hajikhani, A., Tahvanainen, A.-J., Arnold, M., and Suominen, A. Weak signal detection in quantitative foresight: A new method for large-scale data using large language models. In *Technology Mining: Text Analytics for Evidence-Based Foresight*, pp. 99–123. Springer, 2026. doi: 10.1007/978-3-032-10849-4_5. URL https://doi.org/10.1007/978-3-032-10849-4_5.

Bianchi, F., Wang, J., Hasan, Z., Zhu, S., Fourrier, C., and Zou, J. Back to the future: Evaluating AI agents on predicting future events. Hugging Face Blog, 2025. URL https://huggingface.co/blog/futurebench.

Bosse, N. I., Evans, J., Gambee, R. G., Hnyk, D., Mühlbacher, P., Phillips, L., Schwarz, D., and Wildman, J. Deep research bench: Evaluating AI web research agents, 2025. URL https://arxiv.org/abs/2506.06287.

Bosse, N. I., Mühlbacher, P., Wildman, J., Phillips, L., and Schwarz, D. Automating forecasting question generation and resolution for AI evaluation, 2026. URL https://arxiv.org/abs/2601.22444.

Chandak, N., Goel, S., Prabhu, A., Hardt, M., and Geiping, J. Scaling open-ended reasoning to predict the future, 2026. URL https://arxiv.org/abs/2512.25070.

Cheng, P., Liu, J., and Long, Y. PolyBench: Benchmarking LLM forecasting and trading capabilities on live prediction market data, 2026. URL https://arxiv.org/abs/2604.14199.

Diercks, A. M., Katz, J. D., and Wright, J. H. Kalshi and the rise of macro markets. Finance and Economics Discussion Series 2026-010, Board of Governors of the Federal Reserve System, 2026. URL https://doi.org/10.17016/FEDS.2026.010.

Ebadi, A., Auger, A., and Gauthier, Y. Detecting emerging technologies and their evolution using deep learning and weak signal analysis, 2022. URL https://arxiv.org/abs/2205.05449.

Guan, Y., Peng, H., Wang, X., Hou, L., and Li, J. OpenEP: Open-ended future event prediction, 2024. URL https://arxiv.org/abs/2408.06578.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.48550/arXiv.2402.18563. URL https://arxiv.org/abs/2402.18563.

Hansen, A. L., Horton, J. J., Kazinnik, S., Puzzello, D., and Zarifhonarvar, A. Simulating the survey of professional forecasters, 2025. URL https://ssrn.com/abstract=5066286.

Jeen, S., Aitchison, M., and Mantic. Training LLMs to predict world events. Thinking Machines Lab News, 2026. URL https://thinkingmachines.ai/news/training-llms-to-predict-world-events/.

Kalshi. Kalshi research partners with ProCap financial to bring prediction market data to investment research. Kalshi News, 2026. URL https://news.kalshi.com/p/kalshi-procap-partnership-prediction-market-data-investment-research. Accessed: 2026-05-13.

Karger, E., Bastani, H., Yueh-Han, C., Jacobs, Z., Halawi, D., Zhang, F., and Tetlock, P. E. ForecastBench: A dynamic benchmark of AI forecasting capabilities, 2025. URL https://arxiv.org/abs/2409.19839.

Kazinnik, S. and Sinclair, T. M. FOMC In Silico: A multi-agent system for monetary policy decision modeling, 2025. URL https://ssrn.com/abstract=5424097.

Liptay, T., Schwarz, D., Poyiadzi, R., Wildman, J., and Bosse, N. I. Evaluating strategic reasoning in forecasting agents, 2026. URL https://arxiv.org/abs/2604.26106.

Metaculus. FutureEval: Measuring the forecasting accuracy of AI. Metaculus FutureEval, 2026. URL https://www.metaculus.com/futureeval/. Accessed: 2026-05-13.

Mühlbacher, P., Bosse, N. I., and Phillips, L. Towards a realistic long-term benchmark for open-web research agents, 2024. URL https://arxiv.org/abs/2409.14913.

O'Keefe, H. and Marshall, C. Comparing large language models for horizon scanning: A feature assessment. *International Journal of Technology Assessment in Health Care*, 41(S1):S27, 2025. doi: 10.1017/S0266462325101141. URL https://doi.org/10.1017/S0266462325101141.

Patwardhan, T., Dias, R., Proehl, E., Kim, G., Wang, M., Watkins, O., Posada Fishman, S., Aljubeh, M., Thacker, P., Fauconnet, L., Kim, N. S., Miserendino, S., Chabot, G., Li, D., Chao, P., Sharman, M., Barr, A., Glaese, A., and Tworek, J. GDPval: Evaluating AI model performance on real-world economically valuable tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hcuEdq6eKD.

Perigon. Articles api overview. Perigon API Documentation, 2026a. URL https://docs.perigon.io/docs/overview. Accessed: 2026-05-13.

Perigon. Filtering. Perigon API Documentation, 2026b. URL https://docs.perigon.io/docs/search-filters. Accessed: 2026-05-13.

Perigon. Perigon News API: Global media from 80,000 sources. Product page, 2026c. URL https://site.goperigon.com/data-solutions/news-api. Accessed: 2026-05-13.

Perigon. Search concepts. Perigon API Documentation, 2026d. URL https://docs.perigon.io/docs/search-concepts. Accessed: 2026-05-13.

Rozanec, J. M., Sircelj, B., Cochez, M., and Leban, G. Back to the future: Predicting causal relationships influencing oil prices. In *Tiny Papers at the International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TqQlYLd52W.

Schmidt, L., Sharma, O., Marshall, C., and Garcia Gonzalez Moral, S. Horizon scans can be accelerated using novel information retrieval and artificial intelligence tools, 2025. URL https://arxiv.org/abs/2504.01627.

Tao, Z., Wu, P., Jin, Z., Bai, X., Zhao, H., Dou, C., Chen, X., Li, J., Li, L., Tao, C., and Zhang, W. PROPHET: An inferable future forecasting benchmark with causal intervened likelihood estimation, 2025. URL https://arxiv.org/abs/2504.01509.

Turtel, B., Franklin, D., Skotheim, K., Hewitt, L., and Schoenegger, P. Outcome-based reinforcement learning to predict the future, 2025. URL https://arxiv.org/abs/2505.17989.

Turtel, B., Wilczewski, P., Franklin, D., and Skotheim, K. Foresight learning for SEC risk prediction, 2026a. URL https://arxiv.org/abs/2601.19189.

Turtel, B., Wilczewski, P., and Skotheim, K. Forecasting supply chain disruptions with foresight learning, 2026b. URL https://arxiv.org/abs/2604.01298.

Wildman, J., Bosse, N. I., Hnyk, D., Mühlbacher, P., Hambly, F., Evans, J., Schwarz, D., and Phillips, L. Bench to the future: A pastcasting benchmark for forecasting agents, 2025. URL https://arxiv.org/abs/2506.21558.

Wolfers, J. and Zitzewitz, E. Prediction markets. *Journal of Economic Perspectives*, 18(2):107–126, 2004. doi: 10.1257/0895330041371321.

Yang, Q., Mahns, S., Li, S., Gu, A., Wu, J., and Xu, H. LLM-as-a-Prophet: Understanding predictive intelligence with prophet arena, 2025. URL https://arxiv.org/abs/2510.17638.

Ye, C., Hu, Z., Deng, Y., Huang, Z., Ma, M. D., Zhu, Y., and Wang, W. MIRAI: Evaluating LLM agents for event forecasting, 2024. URL https://arxiv.org/abs/2407.01231.

Yuan, Z., Ding, Z., and Vlachos, A. FOReCAst: The future outcome reasoning and confidence assessment benchmark, 2025. URL https://arxiv.org/abs/2502.19676.

Zeng, Z., Liu, J., Chen, S., He, T., Liao, Y., Tian, Y., Wang, J., Wang, Z., Yang, Y., Yin, L., Yin, M., Zhu, Z., Cai, T., Chen, Z., Chen, J., Du, Y., Gao, X., Guo, J., Hu, L., Jiao, J., Li, X., Liu, J., Ni, S., Wen, Z., Zhang, G., Zhang, K., Zhou, X., Blanchet, J., Qiu, X., Wang, M., and Huang, W. FutureX: An advanced live benchmark for LLM agents in future prediction, 2025. URL https://arxiv.org/abs/2508.11987.

Zhang, Z. and Evans, J. Language model perplexity predicts scientific surprise and transformative impact, 2025. URL https://arxiv.org/abs/2509.05591.

Zou, A., Xiao, T., Jia, R., Kwon, J., Mazeika, M., Li, R., Song, D., Steinhardt, J., Evans, O., and Hendrycks, D. Forecasting future world events with neural networks. In *Advances in Neural Information Processing Systems*, 2022. doi: 10.48550/arXiv.2206.15474. URL https://arxiv.org/abs/2206.15474.

A. Code and Data Availability

An anonymized repository containing the benchmark code, prompts and configuration files is available at: https://anonymous.4open.science/r/horizonbench

B. Full Benchmark Pipeline

HorizonBench has three visible parts: constructing a future-side pool of resolved events, generating past-side forecasting questions under a cutoff, and matching generated questions to eligible events. Table 4 summarizes these parts.

The benchmark is target-window anchored: each evaluation run is defined by an area, target window, cutoff date, lead band, and generation model. Table 8 lists the target windows and cutoffs used in the main evaluation.

Future-side event pool construction. For each area, a five-model council searches news from the completed target period and proposes consequential issues that resolved during that period. The council consists of GPT-5.5, Claude Opus 4.7, Gemini 3.1 Pro, Grok 4.3, and DeepSeek v4 Pro. This selection is repeated five times: after each round, the merged list of previously selected events is shown to the council, which is asked to find additional resolved events rather than repeat the same ones. Deduplication is performed in two LLM-based clustering passes. First, after each round, events proposed by the five council members are clustered to merge paraphrases and near-duplicates within that round. Second, after all five rounds, the per-round merged event lists are unioned and clustered again to remove cross-round duplicates. Both clustering passes use GPT-5.5 as the merge model. The resulting canonical event pool is then converted into resolved-event records with canonical uncertainties, resolution metadata, accepted and rejected formulations, and retrospective precursor signals.

Past-side question generation. For each area, target window, cutoff, lead band, and model, the generation agent can search only news published before the cutoff. It produces a ranked list of forecasting-style questions about events that may resolve inside the target window. Generation is also repeated cumulatively: later rounds see the questions already generated in earlier rounds and are asked to add new, non-duplicate questions. In the main metrics, we do not score by round; instead, we truncate the final ranked list at a fixed candidate budget K.

Target windows, cutoffs, and event eligibility. A target window is the future interval whose resolved events are scored. A cutoff is the last date whose news the generation agent can use. A lead band describes the approximate distance between the cutoff and the target window. All date intervals are half-open, $[start, end)$, under UTC start-of-day boundaries. An event is eligible for a matching run if and only if it belongs to the same area and its resolution date falls inside the target window.

Offline matching. For each model-area-window run, the ranked candidate questions and all eligible event records are passed to an offline LLM judge. The judge has no live tools and only determines semantic equivalence between candidate questions and resolved-event records. A match means that the generated question identified the same underlying uncertainty as the event record, not necessarily that it predicted the eventual direction correctly.

C. Event Record Construction and Judge Rubric

Each resolved event is converted into a structured record so that generated questions can be matched to the same underlying uncertainty rather than to exact surface wording. Table 5 summarizes the fields used in each record.

The judge applies polarity-neutral semantic matching. Table 6 gives the condensed decision rubric used to separate true matches from topical near misses.

Record construction. For each selected event, the record-construction agent first verifies the resolving event and date from news sources. It then writes a minimal canonical possibility, usually in the form "Whether [actor] [action] [scope or threshold]." The goal is to avoid binding incidental details, such as codenames, exact weapons, specific sub-actors, or downstream consequences, unless those details are essential to the uncertainty. The agent also writes accepted formulations that should count as the same issue, rejected formulations that should not count, and hindsight signals that appeared before the event resolved.

Polarity-neutral matching. The judge is polarity-neutral. A candidate question can match an event record even if it asks the opposite-polarity version of the same uncertainty. For example, "Will the summit take place?" and "Will the summit be cancelled?" both target the same watchlist issue. The benchmark measures whether the agent surfaced the right issue, not whether it predicted the final YES/NO direction.

Rejected matches. The judge rejects candidates that are only topically related, too broad, or about a neighboring event rather than the same uncertainty. Common false positives include the right actor but wrong action, the right conflict but wrong location, a generic escalation question where the event record requires a specific mechanism, or a candidate that matches one of the record's rejected formulations.

Part	Input	Output	Purpose
Future-side event pool	News from target window	Resolved-event records	Define what later mattered
Past-side generation	News before cutoff	Ranked questions	Test early question discovery
Offline matching	Questions + eligible events	Candidate-event matches	Score coverage and precision

Table 4. HorizonBench pipeline as seen by the evaluation protocol.

Field	Purpose
canonical possibility	Minimal statement of the underlying uncertainty
resolving event	Plain-language description of what resolved
resolved on	Date used for target-window eligibility
accepted formulations	Alternative phrasings that count as the same issue
rejected formulations	Nearby but non-equivalent phrasings
hindsight signals	Retrospective precursor signals for diagnostic analyses

Table 5. Resolved-event record schema.

Case	Judge decision
Same uncertainty, opposite polarity	Match
Same actor, different action	No match
Same topic, too generic	No match
Same event class, wrong geography	No match
Matches rejected formulation	No match

Table 6. Condensed judge rubric.

Target window $[t_s, t_e)$	Cutoff	Lead band
Apr 1–8	Mar 24	d7_14
Apr 8–15	Apr 1	d7_14
Apr 15–22	Apr 8	d7_14
Apr 22–29	Apr 15	d7_14
Apr 1–15	Mar 17	d14_28
Apr 15–29	Apr 1	d14_28

Table 8. Target windows and cutoff dates used in the main evaluation. All dates are in 2026 and all intervals are half-open UTC intervals.

Model	Reasoning setting	Provider route
GPT-5.5	medium	OpenAI direct
Claude Sonnet 4.6	max	Anthropic pinned
DeepSeek v4 Pro	xhigh	DeepSeek pinned
Gemini 3.1 Pro Preview	xhigh	Google AI Studio pinned

Table 9. Generation models and reasoning settings.

D. Experimental Configuration

This section records the concrete benchmark instance used for the main results. A *candidate set* is the merged ranked list of questions produced for one model, one area, and one target window; this is the unit passed to the offline matcher.

Areas and event pools. The evaluation covers two geopolitical areas. Table 7 reports the number of resolved events in each area and the number of candidate sets judged for that area.

Target windows and cutoffs. The main evaluation uses six target windows in April 2026. Four one-week target windows are paired with approximately one-week-ahead cutoffs, and two two-week target windows are paired with approximately two-week-ahead cutoffs. Table 8 lists the half-open UTC windows and their corresponding cutoffs.

Generation models. We evaluate four frontier models. Table 9 records the generation model and reasoning setting used for past-side question generation.

Area	Resolved events	Candidate sets judged
MENA	229	24
EER	216	24
Total	445	48

Table 7. Resolved-event counts and judged candidate sets by area. Each area has 4 generation models $\times$ 6 target windows = 24 candidate sets.

Candidate-budget evaluation. For each model-area-window combination, generation is run in five cumulative passes: later passes see previously generated questions and are asked to add non-duplicate questions. The resulting ranked list is then evaluated at fixed candidate budgets K. This is why the main results report Coverage@K and Precision@K rather than scores indexed by generation pass.

E. News Backend and Cutoff Hygiene

HorizonBench uses Perigon as its historical news backend. Perigon describes its News API as structured news and event data from over 80,000 web sources, enriched with fields such as topics, entities, sentiment, source metadata, and article clustering (Perigon, 2026c). Its article endpoint returns JSON over HTTP and supports search over title, description, and article content, including Boolean operators and exact-phrase matching (Perigon, 2026a;d). We use these features for two purposes: news access during event construction and question generation, and article-count queries for the surprisingness metric.

Cutoff-bound search. The wrapper treats Perigon as a dated evidence source rather than a free-form oracle. During future-side event construction, searches are restricted to the completed target period. During past-side question generation, searches and article reads are restricted to the pre-cutoff period. All date intervals use half-open UTC boundaries, and query end dates are clamped so that the

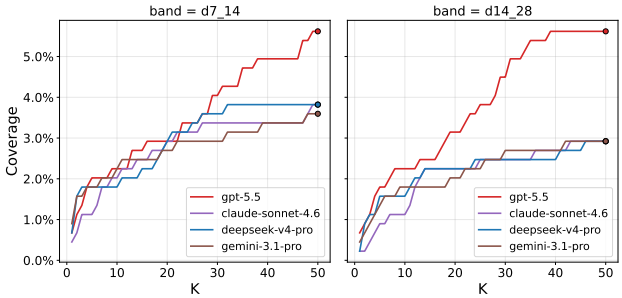

Figure 2. Coverage as a function of candidate budget K. Curves show whether additional ranked question generation continues to recover new future events.

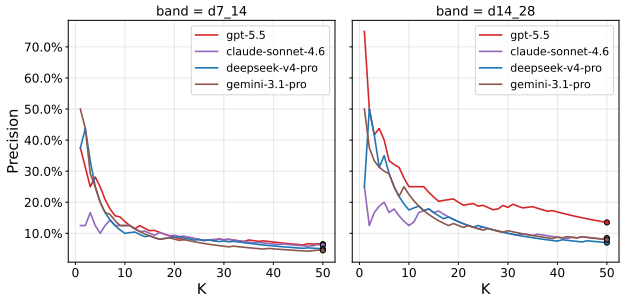

Figure 3. Precision as a function of candidate budget K. Curves quantify the tradeoff between submitting more candidates and maintaining per-candidate quality.

generation agent cannot intentionally request articles after the cutoff. The offline matcher has no Perigon or live-web access.

Surprisingness queries. For the surprisingness analysis, each event is represented by five 3-noun anchor queries. Each anchor set is queried as a Boolean conjunction in matched 7-day pre- and post-resolution windows. The event score is the average smoothed log post/pre article-count ratio across the five anchor queries.

Residual leakage risk. This cutoff hygiene reduces tool-side leakage, but it does not make Perigon a frozen archival snapshot. Residual leakage through article updates, indexing, ranking, clustering, or metadata cannot be fully ruled out; we therefore treat Perigon filtering as an operational safeguard rather than a formal guarantee of archival isolation.

F. Additional Quantitative Results

This section reports quantitative diagnostics that complement the main fixed-budget results. A candidate set is the ranked list of questions generated for one model, one area, and one target window. Unless otherwise stated, Coverage@K uses the 445-event pool as denominator after pooling areas, and Precision@K is candidate-level micro precision among submitted top-K candidates. Precision@K should be read as a labeled-event precision proxy: useful but unrealized forecasting questions are not labeled as correct.

Top-K curves. Table 1 reports three fixed budgets. Figures 2 and 3 show the full curves, which make the inference-time budget tradeoff visible: adding more ranked questions continues to recover some events, but precision falls as lower-ranked candidates are included.

Equal-budget truncation. The K=50 evaluation was run after extending all candidate sets to at least 50 ranked questions and then truncating every candidate set to its first 50 questions. Table 10 records the resulting evaluation size.

Quantity	Value
Candidate sets judged	48
Candidates before truncation	2,827
Candidates after top-50 truncation	2,400
Eligible events	445
MENA events	229
EER events	216

Table 10. Equal-budget K=50 evaluation size. Each candidate set is truncated to exactly 50 ranked questions before matching.

Relationship to GDPval-AA. As a secondary check, we compare HorizonBench performance at $K = 50$ against GDPval-AA Elo, an Artificial Analysis agentic evaluation built on OpenAI's GDPval real-world work-task benchmark (Patwardhan et al., 2026; Artificial Analysis, 2026). Figures 4 and 5 are descriptive only: with four models, they test whether the observed ordering is visibly reducible to an existing benchmark, not whether there is a statistically stable correlation.

Earliest-finder decomposition by area. Table 11 decomposes the earliest-finder result from Table 2 by area. For each event hit by at least one model, credit is assigned to

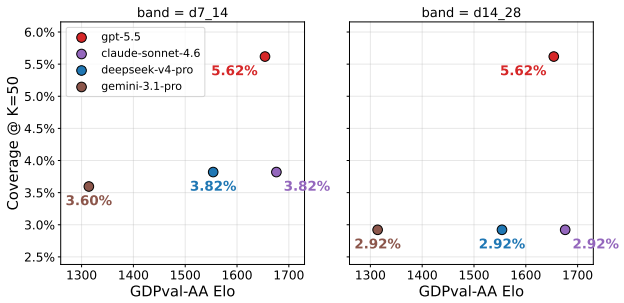

Figure 4. GDPval-AA Elo versus Coverage@50, pooled across areas.

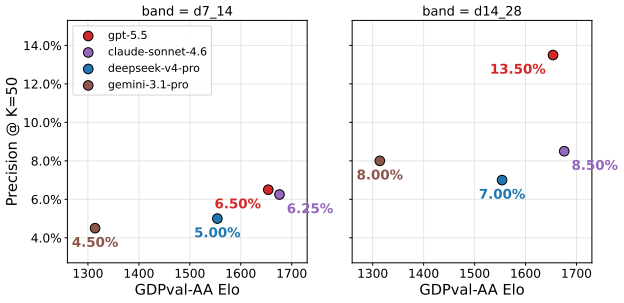

Figure 5. GDPval-AA Elo versus Precision@50, pooled across areas.

Scope	Hit events	GPT-5.5 wins	Claude wins	DeepSeek wins	Gemini wins
MENA	28	12.67 (45.2%)	5.17 (18.5%)	5.33 (19.0%)	4.83 (17.3%)
EER	26	14.75 (56.7%)	4.25 (16.3%)	3.75 (14.4%)	3.25 (12.5%)
Pooled	54	27.42 (50.8%)	9.42 (17.4%)	9.08 (16.8%)	8.08 (15.0%)

Table 11. Per-event-best earliest-finder attribution under the equal-budget top-50 evaluation, with fractional credit for ties.

the model or models that surfaced it at the earliest eligible cutoff, with fractional credit for ties.

G. Qualitative Error Analysis

This section gives representative examples of hits and near misses. The goal is not to introduce another score, but to show what kinds of questions count as successful early discovery and what kinds of topically adjacent questions the judge rejects.

Representative hits. Table 12 shows three successful matches with different levels of panel agreement. They range from an easy scheduled election found by every model to a niche legislative extension found by only one model.

Representative near misses. Table 13 shows three near misses that were semantically close but rejected. These examples illustrate that HorizonBench does not reward generic topical proximity: the generated question must identify the same underlying uncertainty as the resolved-event record.

Manual pattern summary. Table 14 summarizes the three recurring boundary errors illustrated above. These errors are important because they are not simple retrieval failures: in each case, the model reached the right neighborhood of the future event but failed to formulate the precise watchlist issue.

The contrast between Tables 12 and 13 clarifies the benchmark boundary. A successful question need not predict the final polarity, but it must put the same underlying issue on the watchlist. A near miss often names the right actor, region, or broad trend, yet fails because it asks about a neighboring event rather than the event that actually resolved.

H. Prompt and Tool Details

This section describes the prompts and tools used by the agentic parts of HorizonBench. The prompts define the task roles and matching semantics, while the tool wrapper enforces the temporal access constraints programmatically.

Prompt structure. The future-side and past-side prompts are deliberately symmetric. Future-side event selection asks, looking backward after the target window has closed, which consequential uncertainties actually resolved. Past-side question generation asks, looking forward from the cutoff, which consequential uncertainties might resolve in the same target window. Both prompts use the same core constraints: consequentiality, genuine uncertainty, concreteness, diversity, area scope, and target-window alignment. Table 15 summarizes the time perspective, output type, and load-bearing constraints of each prompt.

Event-record prompt. The event-record construction prompt is intentionally stricter than the event-selection prompt. It asks the model to identify the minimal canonical uncertainty, strip incidental details such as codenames or specific platforms, write accepted formulations spanning both polarities, write rejected formulations for nearby but distinct issues, and mine pre-resolution hindsight signals. This record structure is what makes later semantic matching more precise than topical clustering.

Judge prompt. The offline judge compares a ranked candidate list against all eligible event records for the same area and target window. It accepts polarity flips when the underlying uncertainty is the same, but rejects topical near misses. Rejected formulations act as vetoes: if a candidate also matches an explicitly rejected nearby issue, it is not counted as a hit.

Tool surface. Agents interact with Perigon through two news tools: `perigon_search` and `perigon_read_article`. The search tool returns short article records with article IDs, titles, summaries, source domains, and publication dates; URLs are hidden from the model-facing search result to reduce context length. The read tool takes an article ID returned by a previous search and returns the full article text, truncated for model context. Table 16 summarizes which parts of the pipeline have tool access, and Table 17 summarizes the agent-facing tool interface.

Cutoff enforcement. The Perigon wrapper binds the cutoff before tools are exposed to the agent. Agents do not pass or control the cutoff. Search windows are interpreted as half-open UTC intervals $[from, to)$, and any query endpoint after the allowed cutoff or target-period boundary is

Type	Event record	Matched question behavior	Why illustrative
Unanimous hit	**EER-0008: Hungarian parliamentary election.** Canonical issue: whether Viktor Orbán's Fidesz loses its parliamentary majority. Resolved Apr 12, 2026: Tisza won 138/199 seats and Orbán conceded.	All four models matched at rank 1. The accepted questions varied in polarity, e.g., whether Fidesz would retain a majority versus whether Tisza would defeat Fidesz.	Scheduled election with known date, known parties, and a single binary political outcome. This is close to the easiest case for the benchmark.
Partial hit	**EER-0212: UK action against Russian shadow-fleet vessels.** Canonical issue: whether the UK boards or detains at least one Russian shadow-fleet vessel in UK waters. Resolved Apr 28, 2026.	Hit by GPT-5.5 and DeepSeek. GPT-5.5 asked whether UK military or law-enforcement personnel would board, detain, or initiate proceedings; DeepSeek asked whether the UK would forcibly board, seize, or detain a Russian shadow-fleet tanker.	A known policy authorization existed, but the relevant uncertainty was operational follow-through rather than the authorization itself. Half the panel found this boundary.
Sole hit	**EER-0175: Lithuania restrictive measures.** Canonical issue: whether Lithuania extends national restrictions on Russian and Belarusian citizens beyond expiration. Resolved Apr 23, 2026: Seimas voted 95–1 to extend the measures.	Hit only by GPT-5.5, which asked whether Lithuania's Seimas would pass the final vote to extend or amend national restrictive measures during the target window.	Scheduled but niche. The public expiry date was visible, but only one model connected it to the Baltic legislative cycle and the relevant final vote.

Table 12. Representative successful matches. The examples show that hits can range from obvious scheduled political events to niche but forecastable policy follow-through.

Failure type	Event record	Closest rejected candidate	Why rejected
Specificity gap	**EER-0015: Russian drone violates NATO airspace.** Canonical issue: whether a Russian military drone violates the airspace of a NATO member. Resolved Apr 17, 2026, when a Russian attack drone entered Romanian airspace.	DeepSeek asked whether "a Russian military drone or aircraft" would be confirmed to enter NATO airspace.	The added "or aircraft" broadened the question into a rejected neighboring category: manned aircraft incursions are treated as a distinct issue.
Wrong instance	**MENA-0081: Iraq–Syria border crossing.** Canonical issue: whether Iraq and Syria reopen the Rabia/Yarubiyah crossing to the public. Resolved Apr 20, 2026.	Gemini asked whether the Masnaa crossing between Lebanon and Syria would reopen to civilian vehicle traffic.	The model identified the right event class, a Syria-region border reopening, but selected the wrong country pair and crossing.
Adjacent mechanism	**EER-0007: Russia–Ukraine body repatriation.** Canonical issue: whether Ukraine receives at least 1,000 bodies of fallen soldiers in a single exchange with Russia. Resolved Apr 9, 2026.	DeepSeek asked whether Russia and Ukraine would conduct a major prisoner-of-war exchange around Orthodox Easter.	The parent humanitarian-exchange signal was relevant, but living-prisoner exchange and body repatriation are distinct mechanisms and were explicitly separated by the event record.

Table 13. Representative near misses. Each candidate is topically close, but differs from the resolved event on a load-bearing actor, mechanism, or specificity boundary.

Pattern	Example	Observation
Over-broadening	drone OR aircraft	Safety-hedging can collide with explicit rejected formulations.
Right type, wrong instance	Lebanon–Syria vs Iraq–Syria crossing	The event class is correct, but the geographic instance is wrong.
Adjacent mechanism	POW exchange vs body repatriation	The parent process is recognized, but the model chooses the wrong child branch.

Table 14. Manual summary of recurring near-miss patterns.

Prompt	Time perspective	Output	Load-bearing constraints
Event selection	Backward from target-window end	Resolved issues	Consequential, uncertain ex ante, concrete, in-area
Event-record construction	Backward from one resolved issue	Structured event record	Minimal canonical issue, accepted/rejected forms, hindsight signals
Question generation	Forward from cutoff	Ranked candidate questions	Consequential, uncertain, evidence-backed, target-window aligned
Semantic judge	After target window, no tools	Candidate-event matches	Same uncertainty, polarity-neutral, rejected-form veto

Table 15. Prompt roles. The event-selection and question-generation prompts mirror each other around the same target window: one constructs the future-side pool, the other attempts to anticipate it from the cutoff.

Pipeline part	Tool access	Role
Future-side event selection	Perigon search/read	Find resolved issues in target window
Event-record construction	Perigon search/read	Verify events and mine record fields
Past-side question generation	Perigon search/read	Generate cutoff-bounded questions
Offline matching	None	Semantic matching only

Table 16. Tool access by pipeline part. The offline matcher has no Perigon or live-web access.

Tool	Main inputs	Output
`perigon_search`	query, date window, source/language filters	Ranked article summaries with IDs
`perigon_read_article`	article ID from prior search	Full article text and metadata
Submit tools	Structured final payload	Events, records, candidates, or matches

Table 17. Agent-facing tool surface. Article reading is ID-based: agents can only read articles previously returned by search in the same run.

clipped by the wrapper. This design makes temporal filtering a programmatic invariant rather than a prompt-following behavior.

Search syntax and source filters. Perigon search supports Boolean queries over title, description, and article content. The most important practical detail is that Boolean operators must be uppercase: `A AND B` is a conjunction, whereas lowercase `and` is parsed as an ordinary query token. Agents can search either a curated `top100` source group or the broader `all` corpus. The `top100` group is useful for high-precision global coverage, but it is strongly English-centered; local-language coverage generally requires `source_group=all` plus language filters.

Observed tool use in past-side generation. Table 18 reports Perigon tool intensity for past-side question generation in the K=50 pipeline. The unit is one agent run contributing to a candidate set. Models differ substantially in search intensity: Claude issues the most Perigon calls per run, while Gemini uses the fewest.

Source-group behavior. Table 19 shows that models used the news backend differently. GPT-5.5 and Claude relied heavily on the broader `all` corpus, while DeepSeek searched almost exclusively within `top100`. Gemini frequently attempted non-canonical source-group strings such as underscored variants; these calls are grouped as other/invalid.

Model	Search / run	Read / run	Total calls / run
Claude Sonnet 4.6	62.3	19.3	82.6
DeepSeek v4 Pro	39.8	15.2	56.0
GPT-5.5	33.2	13.7	47.9
Gemini 3.1 Pro	16.3	10.2	27.5

Table 18. Average Perigon tool use per past-side generation run. Total calls include search, read, and final submission calls.

Model	top100	all	other / invalid
GPT-5.5	26.1%	73.4%	0.5%
Claude Sonnet 4.6	26.1%	72.3%	1.6%
DeepSeek v4 Pro	98.0%	2.0%	0.0%
Gemini 3.1 Pro	82.4%	5.3%	12.3%

Table 19. Source-group choice across past-side Perigon search calls.

Language use. Table 20 summarizes language-filter behavior. The expected regional languages appear in the search traces: Russian, Ukrainian, and Polish for Eastern Europe / Russia; Arabic and French for Middle East / North Africa. However, model behavior is uneven. GPT-5.5 uses local-language filters most aggressively, while Claude remains almost entirely English-only.

I. Prompt Text

For reproducibility, we include the full prompts used for the four core pipeline parts.

Model	English	Non-English	Common non-English codes
GPT-5.5	59%	41%	ar, ru, uk, fr, pl
Gemini 3.1 Pro	71%	29%	ru, ar, uk, pl, fr, he, ro
DeepSeek v4 Pro	90%	10%	ar, ru, fr, uk, pl
Claude Sonnet 4.6	99.6%	0.4%	ar, fr, es, ru

Table 20. Language-filter usage in past-side Perigon searches. Percentages summarize English versus non-English language filters; searches may include multiple language codes.

Event Selection Prompt

```
I am an admin at Metaculus, and I want to retrospectively measure the quality of the
     Questions submitted to the platform.

The overall process is as follows:

1. At some past date, a large set of Questions was created (researched from news and
     other resources) and posted to Metaculus. Those Questions target uncertainties
    expected to materialize during [{period_start}, {period_end}) -- half-open: the
    start is inclusive, the end is exclusive (UTC start-of-day boundaries).
2. Enumerate the consequential issues that resolved in [{period_start}, {period_end})
    : open uncertainties that should have been on a forecaster's watchlist in advance
    , regardless of which direction they ultimately resolved.
3. Measure what fraction of the step-2 issues were covered by the step-1 Questions:
    how well the question pool anticipated what actually mattered.

Your task is step 2.

You are working in the **geopolitical / military** scope (inter-state conflict,
    diplomatic shifts, alliances, military posture, sanctions, territorial disputes,
    major elections that materially affect foreign-policy posture, government changes
    , energy crises, hybrid attacks, migration shocks, and other consequential
    developments) within the area `{area_name}`.

Your effective "today" is: {period_end}T00:00:00Z (= UTC start-of-day of {period_end
    }; everything published strictly before this moment is in your knowledge). When
    calling `perigon_search`, `to_date={period_end}` returns articles published
    through end of {period_end}-1 day UTC under the half-open `[from_date, to_date)`
    convention -- `to_date` is interpreted as UTC start-of-day, exclusive.
You are looking BACK at the period: [{period_start}, {period_end}) -- half-open UTC
    start-of-day boundaries.

You have access to tools (perigon_search, perigon_read_article) for searching and
    reading news in multiple languages.

English-only searches may miss events primarily covered in local-language press.
    Consider mixing local-language ISO-639-1 codes into `perigon_search` (e.g., `
    language=["en", "fr"]`, `language=["zh", "ja"]`) when relevant to the area.

## Your task

Identify the events that meet ALL of the following. There is no fixed N -- submit as
     many as you can defend:

1. **Consequential**: a serious analyst working in this area would have wanted to
    put this issue on the watchlist BEFORE its outcome was determined. Once the
    resolution materialized (whether affirmative or negative -- happened, didn't
    happen, was cancelled, fell through, etc.), it mattered for decisions, budgets,
    strategy, risk, coordination, markets, policy, or public attention.
2. **Genuinely uncertain in advance**: the outcome was not pre-determined. A
    reasonable forecaster looking at the news weeks earlier would not have given it
    ~100% probability. Surprises that contradicted prior near-certain expectations
    are valid.
3. **Concrete and verifiable**: the issue has a specific date when it resolved,
    identifiable actors, and a clear binary criterion ("did the issue resolve in the
    affirmative or negative" -- either polarity counts as a valid resolution).
4. **Meaningfully diverse across the set**: each event should differ substantively
    from the others. Avoid near-duplicates and slight rephrasings of the same
    underlying event.
5. **Resolution date in the period**: the issue's resolving event (the moment its
    outcome -- affirmative or negative -- became determinate) materialized in [{
    period_start}, {period_end}) -- i.e. on or after {period_start} and strictly
```

```
    before {period_end} UTC start-of-day.
6. **In-area, in-scope**: the event must be primarily about the area `{area_name}`
    and within scope. Cross-area events count if and only if their primary actor or
    resolving locus is inside this area.

## What to avoid

Do not select events that:

- Were essentially pre-determined before resolution: scheduled summits/elections/
    anniversaries with foregone conclusions, announcements that just played out as
    already revealed, etc. (violates condition 2)
- Are too vague to be verified. (violates condition 3)
- Are duplicates or near-duplicates of another event in your set. (violates
    condition 4)
- Are pure restatements of an ongoing trend without a discrete materialization in [{
    period_start}, {period_end}). (violates condition 5)

## Output

When you are done, call `submit_events` exactly once with the events you selected.
    Each event must include:

- `title`: one-line label for the event.
- `resolved_on`: ISO date in [{period_start}, {period_end}) (half-open) when the
    issue's outcome became determinate (affirmative or negative).
- `detail`: 2-3 sentences describing the resolving event (whatever its direction --
    happened, didn't happen, was cancelled, fell through, etc.), who was involved,
    and where. Embed at least one source URL if possible.
- `selection_reason`: 2-3 sentences explaining why this issue was worth putting on a
     forecaster's watchlist in advance.
```

Event-record Prompt

```
I am an admin at Metaculus, and I want to retrospectively measure the quality of the
     Questions submitted to the platform.

The overall process is as follows:

1. At some past date, a large set of Questions was created (researched from news and
     other resources) and posted to Metaculus. Those Questions target uncertainties
    expected to materialize during [{period_start}, {period_end}].
2. Enumerate the consequential issues that resolved in [{period_start}, {period_end
    }]: open uncertainties that should have been on a forecaster's watchlist in
    advance, regardless of which direction they ultimately resolved.
3. Measure what fraction of the step-2 issues were covered by the step-1 Questions:
    how well the question pool anticipated what actually mattered.

The events worth tracking for Step 2 have already been selected. Your task is to
    take ONE such event and write the full benchmark card YAML for it, populating the
     metadata that step 3 will measure coverage against.

You are working in the **geopolitical / military** scope (inter-state conflict,
    diplomatic shifts, alliances, military posture, sanctions, territorial disputes,
    major elections that materially affect foreign-policy posture, government changes
    , energy crises, hybrid attacks, migration shocks, and other consequential
    developments) within the area `{area_name}`.

The target event:
  title: {event_title}
  resolved_on: {event_resolved_on}
```

```
  detail: {event_detail}

Your effective "today" is: {today}T00:00:00Z (= UTC start-of-day of {today};
    everything published strictly before this moment is in your knowledge). When
    calling `perigon_search`, `to_date={today}` returns articles published through
    end of {today}-1 day UTC under the half-open `[from_date, to_date)` convention --
     `to_date` is interpreted as UTC start-of-day, exclusive.
You have access to tools (perigon_search, perigon_read_article) for searching and
    reading news in multiple languages.

English-only searches may miss finer detail in non-English primary coverage.
    Consider mixing local-language ISO-639-1 codes into `perigon_search` (e.g., `
    language=["en", "ar"]`, `language=["fr", "pt"]`) -- local press often carries
    detail on actors, scope, and pre-event signals.

## Methodology (7 steps)

### Step 1: Verify the resolving event

Search for and read authoritative coverage of the event. Confirm: exact `resolved_on`
     date (may differ slightly from the input; use the most reliable source), primary
     actor(s) and target(s), specific resolving development (announcement vs action
    vs aftermath), and a primary URL for `resolution.resolving_source`.

### Step 2: Identify the MINIMAL canonical_possibility (CRITICAL)

The canonical_possibility you will write in Step 3 must be **MINIMAL**. Minimality
    means: every word, every phrase, every modifier, every named entity in the
    canonical sentence must be load-bearing. If any element can be removed (or
    replaced with its broader abstraction) without changing the canonical-issue
    identity, REMOVE / BROADEN it.

Most card-structuring failures come from binding NON-MINIMAL details (specific actor
     configurations, codenames, operation names, named platforms, exact target lists,
     downstream consequences) into canonical_possibility. The agent reads news, sees
    these details prominently mentioned, and assumes they are constitutive. They
    almost never are.

Your job in Step 2 is to identify what the MINIMAL canonical issue is; equivalently,
     the smallest set of features that captures the genuine ex-ante surprise relative
     to pre-event uncertainty.

To find the minimal description, you must run two distinct types of targeted
    searches and triangulate them:

**(A) Pre-event uncertainty discourse**: Search news dated before `resolved_on` (the
     date you verified in Step 1) for background, prior general expectations, and
    forecasts. Use query patterns like:
- "will [event class] happen", "risk of [event class]"
- "[event class] forecast / prediction / analysis"
- "experts say [actor] won't / will [action]"
- "[event class] unlikely / inevitable / could happen"
- contrarian opinion pieces, debate, "could / could not" framings

Read articles to identify the AXIS where pre-event opinion was MOST CONTESTED:
- If pre-event discourse asks "WHETHER [event class] will happen at all" with both
    yes-camp and no-camp publicly arguing, then "did the event class happen at all"
    IS the surprise axis.
- If pre-event discourse takes the event class as given but debates WHO acts, AT
    WHAT SCALE, BY WHAT MECHANISM, then those specifics are the surprise axis.
- If pre-event discourse considered only one specific configuration as inevitable
    and that's what materialized, that specific configuration may be the surprise
    axis (rare; verify carefully).
```

```
**(B) Post-event surprise framing**: Search news at and immediately after `
    resolved_on` (the date you verified in Step 1) to capture how observers reacted.
    Use query patterns like:
- "[event] surprised / shocked / unexpected / dramatic / historic / unprecedented"
- "[event] aftermath / fallout / what does this mean"
- "first time" / "for the first time" + [event class]
- "experts say" / "analysts surprised by" [event aspect]

Read articles to note what specific aspect observers emphasize as NEW, DRAMATIC, or
    UNEXPECTED:
- If reactions emphasize the EVENT HAPPENING ("the [event class] happened despite
    expectations"), then "event class occurred" is the surprise axis.
- If reactions emphasize a SUB-AXIS ("the [margin/scale/threshold] was unprecedented
    "), then that sub-axis is the surprise axis.
- If reactions emphasize an INCIDENTAL DETAIL (codenames, platforms, specific
    personnel) that was operationally newsworthy but predictable conditional on the
    event happening, treat those as INCIDENTAL despite news prominence.

**(C) Triangulate**: the correct surprise axis is the one where (A) pre-event
    discourse showed maximum genuine uncertainty AND (B) post-event reactions framed
    the materialization as surprising. If the two sources diverge, default to the pre
    -event uncertainty axis, because the canonical question must be answerable ex-
    ante before knowing what specifically happened.

After triangulation, apply these three tests to candidate axes:

(a) **Surprise test**: Would an analyst working before `resolved_on` have been
    surprised by this specific detail materializing, or was it predictable
    conditional on the broader event happening?

(b) **Impact test**: Did the world's reaction (markets, policy, public discourse)
    center on this detail, or on a broader fact about the event?

(c) **Counterfactual test**: If this detail had been DIFFERENT but the event had
    still occurred at a higher abstraction level, would observers say "the prediction
     was correct"? If YES, this axis is INCIDENTAL.

Apply these tests to candidate axes such as:

- Whether the event happened at all (usually highest-information axis)
- Timing within the period (often important; sometimes secondary if the period is
    wide)
- Primary actor identity (sometimes important; sometimes any of several plausible
    actors would have counted)
- Actor configuration / coalition composition (often LESS important than the event
    itself)
- Specific scope or threshold (sometimes important if THE THRESHOLD was the surprise)

- Operational details (codenames, platforms, routes, specific personnel): almost
    always INCIDENTAL

**Common failure pattern**: agents bind the SPECIFIC actor configuration that ended
    up materializing into canonical_possibility even when multiple configurations
    were plausible ex-ante. Default to broad scope unless the specific configuration
    was itself the surprise.

Hypothetical illustrations (apply the logic INDEPENDENTLY to YOUR event):

- **A**: A state conducts a kinetic strike against another state's infrastructure.
  - Primary: whether a strike on the target country's sovereign territory occurs at
    all
```

```
  - Secondary: timing within the period; broad target category (military / nuclear /
     civilian / leadership)
  - Incidental: specific codename, weapon platforms, exact attacker-coalition
    composition (if multiple were plausible ex-ante)

- **B**: A long-incumbent ruling party loses a scheduled national election.
  - Primary: whether the incumbent loses its parliamentary majority
  - Secondary: margin of defeat (plurality vs supermajority); peaceful transfer of
    power
  - Incidental: specific final seat counts, post-election coalition arrangements

- **C**: A central bank governor resigns abruptly during a currency crisis.
  - Primary: whether a top central-bank leadership change tied to the crisis
    trajectory occurs
  - Secondary: timing within the crisis; new policy orientation
  - Incidental: specific announcement format, exact resignation date

Identify ONE primary axis where the surprise was concentrated, 0-2 secondary axes
    that materially affect "is this still the same canonical issue" judgments, and
    the incidental axes you will NOT bind into canonical_possibility.

### Step 3: Write canonical_possibility, short_description, and scope_note

`canonical_possibility`: ONE sentence (15-40 words) capturing the primary surprise
    axis **as an open question**. Phrase it as "**Whether** [actor] [action verb] [
    scope/threshold] ..." (or for threshold-type axes, "Whether [observed quantity] [
    exceeds/falls below] [threshold]"). The "whether" form keeps the canonical
    polarity-neutral -- the actual resolution direction lives in `resolution.status`
    and `resolution.resolving_event`, NOT in `canonical_possibility`. Include actor (
    at the abstraction level your Step 2 analysis supports), action verb, and scope/
    threshold (at the surprise-axis level). Do NOT bind incidental details into it.
    Do NOT include calendar date bounds; the judge does not enforce date windows.

**Minimality test (apply BEFORE finalizing; REQUIRED)**: write your draft "Whether
    ..." sentence, then go through it word-by-word, phrase-by-phrase. For each
    modifier, qualifier, named entity, and noun phrase, ask:

- "If I REMOVE this, would the resulting sentence still describe the canonical issue
    ?" If YES, REMOVE the element; it is non-minimal.
- "If I REPLACE this with its BROADER ABSTRACTION (e.g., 'X coordinated with Y' -> '
    X', or 'in Operation Z' -> just the action), would the broader version still
    match what was the ex-ante surprise?" If YES, USE the broader version.

Iterate until every remaining element is load-bearing. Common non-minimal patterns
    to strip out:

- "X (in coordination with Y)" / "X (alongside Y)" -> strip if X-alone variants were
     plausible ex-ante
- "X (in Operation Z)" -> operation names are post-hoc labels; strip
- "X (using platform P)" / "X (with weapon W)" -> platforms / weapons are
    operational details; strip
- "X (causing downstream Y)" -> consequences are not the canonical issue; strip
    unless the canonical IS the consequence
- "X (decisively / overwhelmingly / dramatically)" -> emphasis adverbs are not load-
    bearing; strip
- "X (forming a Y-led government)" / "X (resulting in Z taking power)" -> downstream
     institutional outcomes belong in accepted_formulations, not canonical

Forecaster-correctness test: would a forecaster who correctly identified the
    canonical issue but got a stripped-out detail wrong be considered to have
    predicted CORRECTLY? If YES, the strip is correct. If NO, restore that element.

`short_description`: 2-4 sentences of context plus the resolving event.
```

```
`scope_note`: explicit INCLUSIONS and EXCLUSIONS. INCLUSIONS list phrasings or sub-
    cases all counting as canonical; EXCLUSIONS list nearby events that do NOT count.
     Be explicit; this is the boundary the judge uses to reject near-misses. (
    Abstract shape; INCLUSIONS: any decisive shift in Actor A's posture toward Target
     B regardless of mechanism (sanctions / kinetic / diplomatic). EXCLUSIONS: shifts
     targeting Actor A's proxies outside Target B, or rhetoric without operational
    change.)

### Step 4: Mine hindsight signals

Search pre-`resolved_on` news for >=12 distinct signals that point to the event in
    retrospect. Use varied search queries; do NOT let all signals come from one news
    cluster.

For each signal:
- `date` (<= `resolved_on`)
- `signal` (1-2 sentences of what was reported)
- `significance` (1-3 sentences linking the signal to the resolution via causal
    mechanism; answer "why does this signal point toward the event?"). Abstract shape
    : "Actor X publicly committed Resource R to Mechanism M, raising the precondition
     cost for the resolving event from latent to imminent; the resolution later
    operationalized M directly."
- `source` (a real URL from your Perigon search results; do NOT fabricate URLs or
    invent signals)

### Step 5: Cluster signals into mechanism families

Group your hindsight_signals into named mechanism families. Each family: a short
    noun phrase capturing ONE distinct causal pathway, semantically distinct from the
     others (no overlap). These family names go into `mechanism_families` (Step 7).

### Step 6: Write accepted_formulations and rejected_formulations

`accepted_formulations` (5-9 strings): different forecasting-style yes/no questions,
     all of which would identify the SAME UNDERLYING ISSUE that this card resolves on
    , regardless of which direction it ultimately resolved. Vary wording, time-
    framing, granularity, AND polarity (include both "will X happen?" and "will X be
    cancelled / fail to happen?" framings of the same issue). Capture secondary axes
    from Step 2 here; e.g., one formulation that includes the secondary axis at a
    finer granularity, others that capture only the primary axis.

`rejected_formulations` (5-9 entries each `{{formulation, reason}}`): nearby-but-
    GENUINELY-DISTINCT issue classes that should NOT resolve. Cover: wrong domain,
    wrong actor / wrong target, wrong time / wrong instance, distinct issue class
    with same actors, lower threshold not reached, ambiguous attribution. Do NOT
    include opposite-polarity wording of the same underlying issue here -- those
    belong in `accepted_formulations` (we are measuring whether a forecaster put the
    right ISSUE on the watchlist, not whether they predicted the resolution direction
     correctly).

For each, `reason` is 1-2 sentences explaining why this formulation targets a
    different issue than this card.

**Self-check (CRITICAL; apply before finalizing)**: for each rejected_formulation,
    ask: is this rejected because it's a genuinely-distinct issue class, OR because
    it varies on an INCIDENTAL axis (operational details, codenames, actor sub-
    configurations) OR because it just flips the polarity ("will X happen" vs "will X
     fail / be cancelled" of the SAME underlying issue)? If incidental or polarity-
    only, MOVE it to accepted_formulations. The most common failures here are over-
    rejecting on operational details and over-rejecting on polarity/directionality of
     the same issue.
```

```
### Step 7: Write mechanism_families and lucky_label_match

`mechanism_families`: the list of family names you produced in Step 5. The judge
    will read this list as the card-specific reference cluster for `signal_connected
    `; the generic "real predictive signals" definition is hardcoded judge-side, so
    do not repeat it.

`lucky_label_match` (2-4 sentences): describe what generic-reasoning hits look like
    for THIS card; names that match the right event class but rest only on generic
    reasoning (e.g., "X has been escalating", "Y always conflicts with Z") without
    engaging specific pre-`resolved_on` evidence.

## Output

Call `submit_card` exactly once with:

- `canonical_possibility`: the minimal canonical-issue sentence in "Whether ..."
    form (Step 3); together with `resolution.scope_note`, this defines the boundary
    the downstream judge enforces.
- `short_description`: 2-4 sentences of context plus the resolving event (Step 3).
- `resolution`: `{{status, resolved_on, resolving_event, resolving_source,
    scope_note}}` (Step 1 verifies the resolution; Step 3 writes scope_note). Set `
    status` to "YES" when the canonical_possibility's affirmative materialized (the
    watched action / outcome occurred), or "NO" when the negative materialized (e.g.,
     a planned summit was cancelled, a vote did not pass, an actor declined to act).
    Either polarity is a valid resolution -- what matters is that the underlying
    issue reached determinacy in the period. The `resolving_event` field describes
    what actually happened in plain language regardless of `status`.
- `accepted_formulations`: alternative phrasings the judge recognizes as hits (Step
    6).
- `rejected_formulations`: nearby-but-distinct candidates the judge recognizes as
    misses (Step 6).
- `hindsight_signals`: catalogue of pre-event signals the judge uses as the
    reference cluster for `signal_connected` (Steps 4-5).
- `mechanism_families`: family names for the judge's `signal_connected` rubric (Step
     7).
- `lucky_label_match`: card-specific anti-pattern rubric for `canonical_scope_match`
     (Step 7).

Do not call `submit_card` more than once. The harness fills `card_id` and `subarea`
    post-hoc from the task metadata; you do not write them.
```

Question Generation Prompt

```
I am posting forecasting Questions to Metaculus, and I want them to anticipate
    consequential events well.

The Questions I post will be evaluated as follows:

1. At cutoff {cutoff}, research the world and post a set of Questions to Metaculus.
    These Questions target uncertainties expected to materialize during [{
    target_start}, {target_end}).
2. Enumerate the consequential issues that resolved in [{target_start}, {target_end})
    : open uncertainties that should have been on a forecaster's watchlist in advance
    , regardless of which direction they ultimately resolved.
3. Measure what fraction of the step-2 events were covered by the step-1 Questions:
    how well the question pool anticipated what actually mattered.

Your task is step 1.
```

```
You are working in the **geopolitical / military** scope (inter-state conflict,
    diplomatic shifts, alliances, military posture, sanctions, territorial disputes,
    major elections that materially affect foreign-policy posture, government changes
    , energy crises, hybrid attacks, migration shocks, and other consequential
    developments) within the area `{area_name}`.

Your effective "today" is: {cutoff}T00:00:00Z (= UTC start-of-day of {cutoff};
    everything published strictly before this moment is in your knowledge, everything
     at or after is not).

You are looking FORWARD at the target window: [{target_start}T00:00:00Z, {target_end
    }T00:00:00Z); imagine what could materialize there. Both endpoints are UTC start-
    of-day; the start is inclusive and the end is exclusive (the half-open `[start,
    end)` convention applies uniformly).

You have access to tools (perigon_search, perigon_read_article) for searching and
    reading news in multiple languages. When you call `perigon_search`, the `
    from_date` and `to_date` parameters are bare YYYY-MM-DD strings interpreted as
    UTC start-of-day boundaries -- the same `[from_date, to_date)` half-open
    convention. So `from_date="2026-04-01"` includes articles published from
    `2026-04-01T00:00:00Z` onward, and `to_date="2026-05-01"` returns articles
    published BEFORE `2026-05-01T00:00:00Z` (i.e., articles published through end of
    `2026-04-30` UTC). To search up to right before your effective today, set `
    to_date={cutoff}`.

English-only searches may miss events primarily covered in local-language press.
    Consider mixing local-language ISO-639-1 codes into `perigon_search` (e.g., `
    language=["en", "fr"]`, `language=["zh", "ja"]`) when relevant to the area.

## Your task

Identify candidates that meet ALL of the following:

1. **Consequential**: a serious analyst working in this area would want to put this
    on the table BEFORE its outcome is determined. Once the resolution materializes (
    whether affirmative or negative -- happens, doesn't happen, is cancelled, falls
    through, etc.), it would matter for decisions, budgets, strategy, risk,
    coordination, markets, policy, or public attention.
2. **Genuinely uncertain in advance**: the outcome is not pre-determined. A
    reasonable forecaster would not give it ~100% probability. Low-probability
    surprises that contradict prior near-certain expectations are valid candidates.
3. **Concrete and verifiable**: the candidate has identifiable actors and a clear
    binary criterion ("did the issue resolve in the affirmative or negative" --
    either polarity counts as a valid resolution).
4. **Meaningfully diverse across the set**: each candidate should differ
    substantively from the others. Avoid near-duplicates and slight rephrasings of
    the same underlying scenario.
5. **Resolves within the target window**: expected to resolve approximately within [{
    target_start}, {target_end}).
6. **In-area, in-scope**: the candidate must be primarily about the area `{area_name
    }` and within scope. Cross-area candidates count if and only if their primary
    actor or resolving locus is inside this area.

## What to avoid

Do not submit candidates that:

- Are essentially pre-determined: scheduled summits/elections/anniversaries with
    foregone conclusions, announcements that just played out as already revealed, etc
    . (violates condition 2)
- Are too vague to be verified. (violates condition 3)
- Are duplicates or near-duplicates of another candidate in your set. (violates
    condition 4)
```

```
- Are expected to resolve clearly outside [{target_start}, {target_end}) (e.g. a
    candidate whose resolution can only occur after target_end, or a candidate whose
    resolution must occur before target_start) (violates condition 5)

## Evidence

Every candidate must be tied to specific evidence from the tools. Use the evidence
    field to cite the concrete articles or documents that support the candidate.
    Prefer evidence that points to the actual mechanism behind the candidate, not
    just broad topical similarity.

A modest leap from evidence to candidate is fine; stretch your imagination beyond
    what articles literally say.

## Output

When you are done, call `submit_candidates` exactly once with **a list of distinct
    candidates, ordered from rank 1 (most likely to materialize within [{target_start
    }, {target_end}) AND most consequential if it does) down to the lowest rank still
     meeting the conditions above**.

Each candidate must include:

- `rank`: a 1-based integer matching the candidate's position in the submitted list.
     Number them explicitly: the first candidate gets `rank=1`, the second `rank=2`,
    the third `rank=3`, and so on. The list of `rank` values must form `[1, 2, 3,
    ..., N]` exactly (no gaps, no duplicates).
- `title`: one-line label for the candidate.
- `question`: a forecasting-style yes/no question or short hypothesis. Prefer
    wording that is specific enough to be meaningful, but do not force artificial
    precision if the underlying uncertainty is still early and fuzzy.
- `rationale`: 4-6 sentences explaining why the candidate is worth asking about.
- `evidence`: list of concrete articles or documents that most directly support the
    candidate.

You are not required to produce fully specified resolution criteria. What matters is
     what evidence visible at {cutoff} supports taking the candidate seriously.

**There is no fixed N. Submit as many as you can defend, but each one must clear all
     conditions above. Do not pad the list with weak candidates; the evaluation
    rewards covering real events, not list length.**


## How to work

- Search broadly, then follow promising threads. Look across the major sub-themes
    inside the area + scope.
- When a thread looks potentially important for the target interval, investigate it
    further using additional searches and article reads.
- Look for both:
  - direct continuations of the current situation, and
  - second-order or cross-domain consequences that may become important by the
    target interval.
- Actively try to find candidates that are easy to miss at {cutoff}, not just the
    most obvious headline continuation.
```

Semantic Judge Prompt

```
I am an admin at Metaculus, and I want to retrospectively measure the quality of the
     Questions submitted to the platform.
```

```
The overall process is as follows:

1. At some past date, a large set of Questions was created (researched from news and
     other resources) and posted to Metaculus. Those Questions target uncertainties
    expected to materialize during [{target_start}, {target_end}) -- half-open: the
    start is inclusive, the end is exclusive (UTC start-of-day boundaries).
2. Enumerate the consequential issues that resolved in [{target_start}, {target_end})
    : open uncertainties that should have been on a forecaster's watchlist in advance
    .
3. Measure what fraction of the step-2 issues were covered by the step-1 Questions:
    how well the question pool anticipated what actually mattered.

Your task is step 3.

You are working in the **geopolitical / military** scope (inter-state conflict,
    diplomatic shifts, alliances, military posture, sanctions, territorial disputes,
    major elections that materially affect foreign-policy posture, government changes
    , energy crises, hybrid attacks, migration shocks, and other consequential
    developments) within the area `{area_name}`.

Your effective "today" is {target_end}.
You are looking BACK at the period: [{target_start}, {target_end}) -- half-open UTC
    start-of-day boundaries.

## Your task

You will be given:

- A **list of N candidate forecasting questions** submitted at step 1. Each
    candidate has:
  - `question` (a yes/no question or short hypothesis)
  - `rationale` (4-6 sentences explaining why the candidate was worth asking about,
    typically with dated signals from the news)
- A **pool of "truth" cards** -- the actual issues that resolved during [{
    target_start}, {target_end}) from step 2. Each card has:
  - `card_id` (string identifier -- use this in your output)
  - `resolved_on` (ISO date the resolving event occurred)
  - `canonical_possibility` (one sentence naming actor + action + threshold)
  - `short_description` (2-4 sentences of context + the resolving event)
  - `resolving_event` (1-2 sentences naming actor / action / target -- the actual
    outcome, whether the affirmative or negative materialized)
  - `scope_note` (INCLUSIONS / EXCLUSIONS text -- variants that all count as
    canonical, and nearby events that do NOT count)
  - `accepted_formulations` (5-9 phrasings spanning either polarity, all identifying
     the same underlying issue this card resolves on)
  - `rejected_formulations` (5-9 genuinely-distinct issue classes with reasons;
    opposite-polarity wording of the same issue is intentionally NOT here -- those
    live in `accepted_formulations`)
  - `lucky_label_match` (optional 2-4 sentence anti-pattern: generic-prior hits that
     name the right event class but engage no specific evidence)

For each (candidate, card) pair, decide whether they match. A pair **matches** iff
    ALL of the following hold:

1. **Semantic alignment to canonical**: the candidate question's actor + action +
    scope/threshold corresponds to the card's `canonical_possibility` or to any of
    its `accepted_formulations`. Wording need not be literal -- paraphrases,
    alternative time-framings within the period, and slightly different specific
    instances all count, as long as the underlying scenario is the same. Polarity (
    affirmative vs negative wording) is irrelevant to alignment -- the issue is what
    matters, not which direction the candidate predicted.
2. **No rejected match**: the candidate does NOT semantically match any of the card'
    s `rejected_formulations`. The rejected list is exhaustive of explicit exclusions
```

```
    ; if a candidate fits BOTH an accepted_formulation AND a rejected_formulation,
    the rejected one wins -> not a match.
3. **Not a generic-prior "lucky label"**: the candidate is not a phrase that names
    the right actor / event class but rests only on broad priors without engaging the
     specific scenario named by the card. If the card supplies a `lucky_label_match`
    anti-pattern, apply it.

A single candidate may match multiple cards (rare -- happens when two cards capture
    related-but-distinct events). A single card may match multiple candidates (the
    agent submitted variants).

Walk through every candidate and check it against every card; then sweep cards that
    received zero matches to catch wording-variant misses.

Note: this judgment does NOT evaluate whether the candidate's evidence is signal-
    connected to the card's pre-cutoff signal cluster. That axis is intentionally out
     of scope here.

## What to reject

**Polarity is NOT a reject reason.** The benchmark measures whether the right
    underlying ISSUE was on the watchlist, not whether the direction was predicted
    correctly. Match against `canonical_possibility` / `accepted_formulations` -- the
     issue -- not against the specific direction `resolving_event` took. For example:
     both "Will summit happen?" and "Will summit be cancelled?" match a card whose
    underlying issue is summit happen/cancel, regardless of which polarity ultimately
     materialized.

Do NOT record a match when:

- The candidate shares only the actor or only the event-class with the card, but the
     proposed specific outcome differs (right neighborhood, wrong scenario).
- The candidate matches any `rejected_formulation` of the card -- that veto stands
    even if an `accepted_formulation` also matches.
- The card's `canonical_possibility` requires a JOINT occurrence ("X AND Y") and the
     candidate asks only about ONE of the conjuncts. The candidate must cover the
    joint event to match.
- The candidate is a generic-prior label hit ("Iran tensions escalate", "ME peace
    deal stalls") without naming the specific scenario the card resolves on.

The benchmark's design penalizes near-miss false positives more than misses, so when
     in doubt, do not record a match.

## Output

Submit your verdict via the `submit_window_batch_verdict` tool call, exactly once.
    The tool expects:

  {{
    "matches": [
      {{
        "candidate_rank": <1-based int>,
        "card_id": "<card_id from the cards section>",
        "rationale": "<1-2 sentences naming which accepted_formulation the candidate
     aligns with, and why no rejected_formulation applies>"
      }},
      ... one entry per (candidate, card) match ...
    ],
    "summary_rationale": "<2-4 sentences summarizing the verdict across all
    candidates and cards in scope>"
  }}
```

```
Only list pairs that match. Do NOT list non-matches. Do NOT output any free text
    outside the tool call. Do NOT call any other tool.
```